I CLAIM:

1. A method for extending an extensible framework comprising the steps of:

providing the extensible framework;

providing an extension module for the extensible framework; and

providing an abstraction layer overlaying the extensible framework, the abstraction layer including a uniform cross-platform interface between the extension module and the extensible framework.

2. A method for extending an extensible framework according to Claim 1, wherein the abstraction layer is compatible with a second extensible framework and the uniform cross-platform interface is framework-independent.

3. A method for extending a first extensible ordered data flow-based framework comprising the steps of:

providing the first extensible ordered data flow-based framework;

providing an extension module for the first extensible ordered data flow-based framework;

providing an abstraction layer overlaying the first extensible ordered data flow-based framework, the abstraction

layer including a uniform cross-platform interface between the extension module and the first extensible ordered data flow-based framework; and

synchronizing the abstraction layer to data flow in the first extensible ordered data flow-based framework.

4. A method for extending a first extensible ordered data flow-based framework according to Claim 3, wherein data of the data flow is arranged in time progression.

5. A method for extending a first extensible ordered data flow-based framework according to Claim 4, wherein the first extensible ordered data flow-based framework is a multimedia presentation application, the method further comprising the step of rendering data-types included in the data.

6. A method for extending a first extensible ordered data flow-based framework according to Claim 4, wherein said synchronizing step includes the step of the first extensible ordered data flow-based framework calling the extension module with current time.

7. A method for extending a first extensible ordered data flow-based framework according to Claim 4, wherein said

synchronizing step includes the step of the extension module calling the first extensible ordered data flow-based framework with current time.

8. A method for extending a first extensible ordered data flow-based framework according to Claim 3, wherein data of the data flow is ordered in multiple dimensions.

9. A method for extending a first extensible ordered data flow-based framework according to Claim 3, further including the step of the first extensible ordered data flow-based framework calling the extension module with current position of data.

10. A method for extending a first extensible ordered data flow-based framework according to Claim 3, further including the step of the extension module calling the first extensible ordered data flow-based framework with current position of data.

11. A method for extending a first extensible ordered data flow-based framework according to Claim 3, further including the step of the first extensible ordered data flow-based framework calling the extension module with current rate of flow of data.

12. A method for extending a first extensible ordered data flow-based framework according to Claim 3, further including the step of the extension module calling the first extensible ordered data flow-based framework with current rate of flow of data.

13. A method for extending a first extensible ordered data flow-based framework according to Claim 3, further including the step of the first extensible ordered data flow-based framework calling the extension module with current direction of flow of data.

14. A method for extending a first extensible ordered data flow-based framework according to Claim 3, further including the step of the extension module calling the first extensible ordered data flow-based framework with current direction of flow of data.

15. A method for extending a first extensible ordered data flow-based framework according to Claim 3, wherein data of the data flow represents discrete states.

16. A method for extending a first extensible ordered data flow-based framework according to any one of Claims 3-15, wherein the abstraction layer includes a mechanism for executing platform-independent code.

17. A method for extending a first extensible ordered data flow-based framework according to any one of Claims 3, 5, 6, or 7, wherein data of the data flow is packetized and transported to the first extensible ordered data flow-based framework over a network, and the abstraction layer includes a mechanism for executing platform-independent code.

18. A method for extending a first extensible ordered data flow-based framework according to any one of Claims 4, or 8-15, wherein the data of the data flow is packetized and transported to the first extensible ordered data flow-based framework over a network, and the abstraction layer includes a mechanism for executing platform-independent code.

19. A method for extending a first extensible ordered data flow-based framework according to any one of Claims 3-15, wherein the abstraction layer includes a Java™ Virtual Machine.

20. A method for extending a first extensible ordered data flow-based framework according to any one of Claims 3, 5, 6, or 7, wherein data of the data flow is packetized and transported to the first extensible ordered data flow-based framework over a network, and the abstraction layer includes a Java™ Virtual Machine.

21. A method for extending a first extensible ordered data flow-based framework according to any one of Claims 4, or 8-15, wherein the data of the data flow is packetized and transported to the first extensible ordered data flow-based framework from a server storing the data over a network, and the abstraction layer includes a Java™ Virtual Machine.

22. A method for extending a first extensible ordered data flow-based framework according to Claim 21, wherein the first extensible ordered data flow-based framework is a RealPlayer.

23. A method for extending a first extensible ordered data flow-based framework according to any one of Claims 3-15, wherein the abstraction layer is compatible with a second extensible ordered data flow-based framework and the uniform cross-platform interface is framework-independent.

24. A method for extending an extensible ordered data flow-based framework comprising the steps of:

providing the extensible ordered data flow-based framework;

providing an extension module for the extensible ordered data flow-based framework;

providing an abstraction layer overlaying the extensible ordered data flow-based framework, the abstraction layer

including means for uniform cross-platform framework-independent interconnection between the extension module and the extensible ordered data flow-based framework; and

step for synchronizing the abstraction layer to data flow in the extensible ordered data flow-based framework.

25. Apparatus for using an ordered file through applications including an extensible ordered data flow-based framework, an extension module, and an abstraction layer having a uniform cross-platform interface between the extensible ordered data flow-based framework and the extension module, the apparatus comprising:

a computing machine running the extensible ordered data flow-based framework, the extension module, and the abstraction layer;

means for synchronizing the extension module and the extensible ordered data flow-based framework;

means for providing content of the ordered file to the extensible ordered data flow-based framework.

26. Apparatus for using an ordered file according to Claim 25, further including means for rendering data-types included in the ordered file.

27. Apparatus for using an ordered file according to Claim 25, wherein the means for providing content includes a network connection and a first server storing the ordered file.